Highlights

Symbolic Kolmogorov-Arnold Networks as a Constant-Time Alternative to Linear Time-Varying Model Predictive Control for Embedded Control of Non-Minimum Phase Systems

- Symbolic KANs approximate LTV-MPC policies for non-minimum phase systems.
- KANs reduce inference latency by up to five orders of magnitude over MPC baselines.
- Symbolic extraction provides mathematical transparency and safety verification.
- Microsecond execution enables optimal control deployment on low-cost edge microcontrollers.

Symbolic Kolmogorov-Arnold Networks as a Constant-Time Alternative to Linear Time-Varying Model Predictive Control for Embedded Control of Non-Minimum Phase Systems[⋆,⋆⋆]

ARTICLE INFO

Keywords:
Kolmogorov-Arnold networks
model predictive control
real-time control
interpretable neural networks
embedded control systems
non-minimum phase systems

ABSTRACT

Model Predictive Control (MPC) is the gold standard for constrained multi-variable control, yet its computational burden often precludes deployment on low-cost embedded hardware. As a primary contribution to artificial intelligence, this work proposes distilling complex control policies into a symbolic Kolmogorov-Arnold network (KAN), creating a computationally efficient, explicit neural controller. For the engineering application, both approaches were benchmarked on a canonical Johansson quadruple-tank system exhibiting severe non-minimum phase (NMP) dynamics. Experiments conducted via hardware-in-the-loop (HIL) demonstrate that the symbolic KAN approximates the MPC policy with high fidelity while reducing inference time by up to five orders of magnitude (from 19ms with stochastic jitter down to 1µs). This effectively solves the hardware limitations of edge devices, enabling advanced control on standard industrial microcontrollers.

1. Introduction

Control of non-minimum phase (NMP) systems remains a significant challenge due to their inherent inverse response dynamics (Skogestad and Postlethwaite, 2005). Characterized by the presence of a non-minimum phase zero in the right half-plane, these systems exhibit initial trajectories that move in the opposite direction of the steady-state target. Standard reactive controllers, such as PID, often induce instability when aggressively tuned for such dynamics. Consequently, Model Predictive Control (MPC) has become the industrial standard for NMP systems. By optimizing a cost function over a receding horizon, MPC provides a principled framework to handle multivariate constraints while offering theoretical guarantees on closed-loop stability and optimality (Borrelli et al., 2017; Mayne et al., 2000). However, the requirement for solving quadratic programming (QP) problems online creates a severe computational bottleneck, often precluding deployment on low-cost, resource-constrained embedded edge devices.

While Bemporad (Bemporad et al., 2002) proposed explicit MPC to shift the computational burden offline via multi-parametric programming, this approach scales poorly. The exponential growth of the polyhedral partition complexity with respect to the prediction horizon and state dimensions creates a prohibitive memory footprint. For instance, Alvarado et al. (Alvarado et al., 2011) successfully implemented explicit MPC on a quadruple-tank system, but were constrained to short prediction horizons and low sampling frequencies due to the hardware limitations of the era and the vast complexity of the parametric solution.

While modern primal-dual solvers (Stellato et al., 2020; Bambade et al., 2025; Jordana et al., 2023) mitigate online solution times compared to earlier active-set methods, they remain fundamentally iterative. Depending on active set changes and model mismatches, these solvers exhibit non-deterministic execution times. In high-frequency embedded contexts, this temporal variance creates stochastic latency spikes, violating the strict real-time deadlines required for safe cyber-physical operation.

To circumvent the computational overhead of iterative solvers, recent advances in scientific machine learning propose distilling heavy control policies into neural networks (Hewing et al., 2020; Adhau et al., 2024). Promising frameworks for such data-driven control have been introduced by Raissi (Raissi, 2018) via physics-informed neural networks, Brunton et al. (Brunton et al., 2016) via sparse identification, and Chen et al. (Chen et al., 2018) through neural ordinary differential equations, among others (Putri et al., 2024; Ramadevi et al., 2023; Zhang et al., 2022; Lopez et al., 2022). However, a critical limitation persists across these methodologies: the reliance on standard Multi-Layer Perceptrons (MLPs). While the universal approximation theorem guarantees that MLPs can replicate MPC policies, their dense matrices of fixed activation functions operate as impenetrable "black boxes" (Rudin, 2019). This lack

ORCID(s):

of mathematical interpretability makes it impossible to formally verify absence of positive feedback loops or latent instabilities, rendering standard neural controllers highly hazardous for safety-critical industrial applications (Amodei et al., 2016).

This paper introduces a methodology using Kolmogorov-Arnold networks (KANs) (Liu et al., 2024) to bridge the gap between advanced control and embedded safety. Unlike MLPs, which apply fixed activation functions at the nodes, KANs learn smooth, spline-based activation functions along the edges of the network. By exploiting this structure, the network can be compressed via symbolic regression into a tractable, explicit mathematical equation.

In this work, a symbolic KAN is proposed as a computationally efficient explicit controller for a multivariable quadruple-tank process exhibiting severe NMP dynamics. By extracting a fully auditable control law, the predictive intelligence of Linear Time-Varying (LTV) MPC is preserved while condensing inference to constant-time polynomial evaluations.

The primary contributions of this paper are multifold: a gain-scheduled KAN architecture is trained to accurately approximate iterative LTV-MPC policies across both minimum and non-minimum phase regimes; a methodology for extracting the KAN into a raw symbolic format is demonstrated. This enables a systematic, scalable Jacobian evaluation ($\frac{\partial u_k}{\partial e_k} < 0$, where u_k and e_k describe the control law and setpoint deviation respectively) to formally detect and eliminate unsafe positive feedback topologies prior to C deployment. Through hardware-in-the-loop (HIL) validation on an STM32H7 microcontroller, the symbolic control law is shown to reduce memory footprint by an order of magnitude and inference latency by up to five orders of magnitude ($O(1)$ execution at $1\mu s$) compared to primal-dual solvers. The neuro-symbolic controller is rigorously validated against structural boundary handovers, unmodeled actuator degradation, and previously unseen reference trajectories, demonstrating robust asymptotic convergence without online retraining.

The remainder of this paper is organized as follows: Section II outlines the problem formulation of this article. Section III describes the mathematical modeling of the NMP quadruple-tank benchmark. Section IV details the control architecture, the LTV-MPC formulation, and the KAN extraction process. Section IV presents the embedded implementation and empirical hardware-in-the-loop results. Finally, Section V discusses the implications for deterministic edge AI, followed by conclusions and future work.

2. Problem Formulation

2.1. The Kolmogorov-Arnold Representation Theorem

Unlike universal approximation theorem standing at the foundation of MLPs, Kolmogorov-Arnold representation theorem (Kolmogorov, 1957; Arnold, 1957) states that any multivariate function $f(x)$ can be represented as a finite composition of continuous univariate functions:

$$f(x) = \sum_{q=1}^{2n+1} \Phi_q\left(\sum_{p=1}^{q} \phi_{q,p}(x_p)\right) \tag{1}$$

where Φ_q and $\phi_{q,p}(x_p)$ represent the univariate functions representing the finite composition.

While standard MLPs place fixed activation functions at the nodes and learn linear weights, the KAN architecture parametrizes the univariate functions $\phi_{q,p}$ and Φ_q as learnable B-splines on the edges (Liu et al., 2024). This structural shift allows the network to be explicitly distilled into the symbolic polynomials required for deterministic embedded control.

2.2. Control Objective

The control objective is to regulate the liquid levels of a nonlinear multivariable quadruple-tank process to prescribed reference trajectories under input saturation, measurement noise, and adversarial actuator uncertainty. The system dynamics are governed by Bernoulli's principle and Torricelli's law, yielding a strongly coupled, non-minimum phase (NMP) nonlinear state-space model (Johansson, 2000).

Let $x \in \mathbb{R}^4$ denote the tank levels and $u \in \mathbb{R}^2$ the pump voltages. The control task is to minimize level tracking error subject to physical input constraints while maintaining closed-loop stability.

While Model Predictive Control (MPC) provides a principled framework for handling constraints, its real-time feasibility in embedded settings is limited by repeated quadratic program solution, accurate model availability, and estimator–controller coupling (Mayne et al., 2000; Borrelli et al., 2017). In NMP configurations with plant drift or actuator degradation, these requirements may be violated.

This work considers an alternative formulation in which an explicit nonlinear control policy is learned offline and executed online with fixed computational complexity, avoiding optimization at runtime while preserving stabilizing behavior.

3. Mathematical Modeling

The system under test is the canonical quadruple-tank benchmark — a nonlinear multivariable benchmark consisting of four interconnected liquid tanks and two control inputs, exhibiting strong cross-coupling and NMP behavior (Johansson, 2000). The model of the system is described by following equations:

$$\begin{aligned}
\frac{dh_1}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1}u_1 \\
\frac{dh_2}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2}u_2 \\
\frac{dh_3}{dt} &= -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2)k_2}{A_3}u_2 \\
\frac{dh_4}{dt} &= -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1)k_1}{A_4}u_1
\end{aligned} \tag{2}$$

The plant diagram is provided in Fig. 1. Table 1 provides the physical values used in the simulation. In (2), $g = 981\frac{\text{cm}}{\text{s}^2}$; $h_i,\ i \in [1,4]$ represent water heights in the tanks from 1 to 4; $u_k,\ k \in [1,2]$ denotes the control inputs. Table 1 shows the experimental parameters.

Table 1
Experimental Parameters for MP and NMP Configurations

Parameter	Symbol	MP	NMP
Valve Ratios	γ_1, γ_2	$0.70, 0.60$	$0.43, 0.34$
Pump Gains ($\frac{\text{cm}^3}{Vs}$)	k_1, k_2	$3.33, 3.35$	$3.14, 3.29$
Outlet cross-section area (cm^2)	a_i	$[0.071, 0.057, 0.071, 0.057]^T$	
Tank cross-section area (cm^2)	A_i	$[28, 32, 28, 32]^T$	
Initial State (cm)	x_0	$[12.4, 12.7, 1.8, 1.4]^T$	$[12.6, 13.0, 4.8, 4.9]^T$
Target State (cm)	x_{ref}	$[10.0, 10.0, 2.0, 2.0]^T$	

To tackle the nonlinear dynamics within a convex optimization framework (Boyd and Vandenberghe, 2004), successive linearization is deployed. At each time step k, the nonlinear model (1) is approximated by an Affine Linear Time-Varying (LTV) structure:

$$\begin{aligned}
x_{k+1} &= A_k x_k + B_k u_k + d_k \\
y_k &= C x_k
\end{aligned} \tag{3}$$

where d_k represents the linearization offset. The Jacobian matrix A is defined as:

$$A = \begin{pmatrix}
-\frac{a_1}{A_1}\sqrt{\frac{g}{2h_1}} & 0 & \frac{a_3}{A_1}\sqrt{\frac{g}{2h_3}} & 0 \\
0 & -\frac{a_2}{A_2}\sqrt{\frac{g}{2h_2}} & 0 & \frac{a_4}{A_2}\sqrt{\frac{g}{2h_4}} \\
0 & 0 & -\frac{a_3}{A_3}\sqrt{\frac{g}{2h_3}} & 0 \\
0 & 0 & 0 & -\frac{a_4}{A_4}\sqrt{\frac{g}{2h_4}}
\end{pmatrix} \tag{4}$$

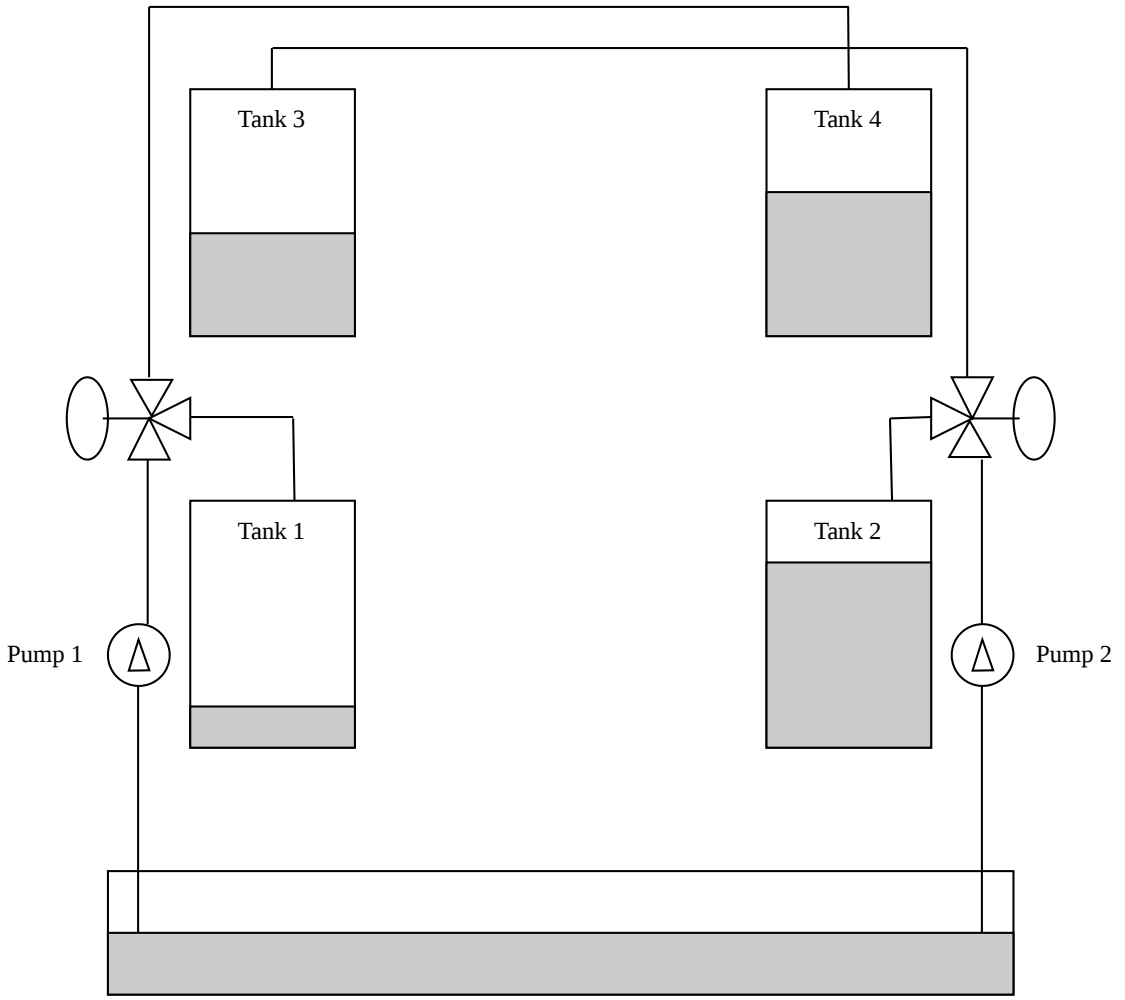

Figure 1: Quadruple-tank process. As described, this plant features four interconnected tanks and two pumps and exhibits strong cross-coupling. A nontrivial approach is required for NMP configuration of the plant.

Matrix $\boldsymbol{B}$ is given by Johansson as:

$$\boldsymbol{B} = \begin{pmatrix} \frac{\gamma_1 k_1}{A_1} & 0 \\ 0 & \frac{\gamma_2 k_2}{A_2} \\ 0 & \frac{(1-\gamma_2) k_2}{A_3} \\ \frac{(1-\gamma_1) k_1}{A_4} & 0 \end{pmatrix} \tag{5}$$

Matrix $\boldsymbol{C}$ is defined as:

$$\boldsymbol{C} = \begin{pmatrix} k_{11} & 0 & 0 & 0 \\ 0 & k_{22} & 0 & 0 \\ 0 & 0 & k_{33} & 0 \\ 0 & 0 & 0 & k_{44} \end{pmatrix} \tag{6}$$

where $k_{ii},\ i \in [1, 4]$ had been set to 1 $\frac{V}{cm}$ for direct conversion.

Although the HIL setup utilizes the full measurement matrix in (6), the underlying system's structural observability was verified for the standard partial output configuration ($y = [h_1, h_2]^T$), satisfying Kalman rank controllability and observability conditions consistent with (Johansson, 2000).

4. Control Architecture

The control loop consists of canonical Extended Kalman Filter (EKF) (Kalman, 1960) for state estimation and a Gain-Scheduled Controller (LTV-MPC/symbolic KAN).

4.1. Stochastic Estimation (EKF)

While the standard Quadruple-Tank process typically relies on partial observability ($y = [h_1, h_2]^T$), for this HIL benchmark full state availability ($y = x$) was assumed to decouple the control law performance from observer convergence dynamics. The EKF was implemented with an identity measurement matrix ($H = I_4$) to filter Gaussian sensor noise injected by the plant simulation, ensuring that both KAN and MPC operated on comparable, clean state estimates.

4.2. LTV-MPC Formulation

The optimal control problem was first solved using the CVXPY (Diamond and Boyd, 2016) extension for Python, which was then converted via CVXpygen framework to the Operator Splitting Quadratic Problem (OSQP) solver (Stellato et al., 2020), a successor of CVXGEN (Mattingley and Boyd, 2012). The cost function minimizes tracking error and control effort:

$$J = \sum_{k=0}^{N-1} (\|x_k - x_{ref}\|^2 Q + \|u_k\|^2 R) \tag{7}$$

In matrix form, this equation becomes:

$$J = \sum_{k=0}^{N-1} (\|x_k - x_{ref}\|^T Q \|x_k - x_{ref}\| + \|u_k\|^T R \|u_k\|) \tag{8}$$

The tuning of the LTV-MPC follows the state-input tuning guidelines established in literature (Mayne et al., 2000; Borrelli et al., 2017). Matrix Q was tuned as:

$$Q = \begin{pmatrix} 40 & 0 & 0 & 0 \\ 0 & 40 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{pmatrix}$$

Matrix R was tuned as:

$$R = \begin{pmatrix} 0.001 & 0 \\ 0 & 0.001 \end{pmatrix}$$

The structure followed the Linear Time-Varying methodology, re-linearizing the plant every discrete time interval. The prediction horizon is N = 30.

The model was re-linearized at each sampling instant about the current state estimate $\hat{x}_k$ using Euler discretization with sampling time $dt = 0.003s$, while the prediction model was discretized with interval $\Delta t_p = 0.1s$ to reduce optimization complexity.

Additionally, an assumption is made:

Assumption 1: The reference trajectory x_{ref} is assumed to be bounded and piecewise continuously differentiable. This smoothness condition ensures that the required control effort remains within the physical saturation limits of the actuators, mitigating unbounded transient spikes inherent to the inverse response of NMP dynamics.

4.3. The Student: Symbolic Kolmogorov-Arnold Network

Two symbolic KANs are structured to approximate the optimal control law $u_k = \pi(x_{est}, x_{ref})$, where π denotes the mapping between the height estimated by the EKF (x_{est}) and the reference height (x_{ref}). To accommodate the distinct dynamics of the NMP and minimum phase (MP) regimes, a Gain-Scheduled architecture was deployed; the active network is switched based on valve ratios γ: $\gamma_1 + \gamma_2 > 1$ triggers MP and $0 < \gamma_1 + \gamma_2 < 1$ triggers NMP.

A symbolic Kolmogorov–Arnold Network (KAN) is trained offline to approximate the nonlinear MPC feedback law using L-BFGS optimization algorithm (Wright et al., 1999) over stochastic gradient descent (Robbins and Monro, 1951). The network settings have been initialized as [8, 5, 2], where 8 represents the input dimensions, corresponding to the heights of the four tanks and four deviations from the target state, 5 represents the hidden neurons and 2 represents the outputs, i.e. pump voltages. The KAN architecture employs fixed univariate basis functions with a shallow topology, enabling symbolic interpretability and deterministic execution. Mean-squared error loss is minimized using batch least squares optimization (Legendre, 1805). Once trained, the resulting symbolic control law is deployed directly on the target platform without online optimization or gradient-based updates.

The neural network's structure is outlined in Fig. 2.

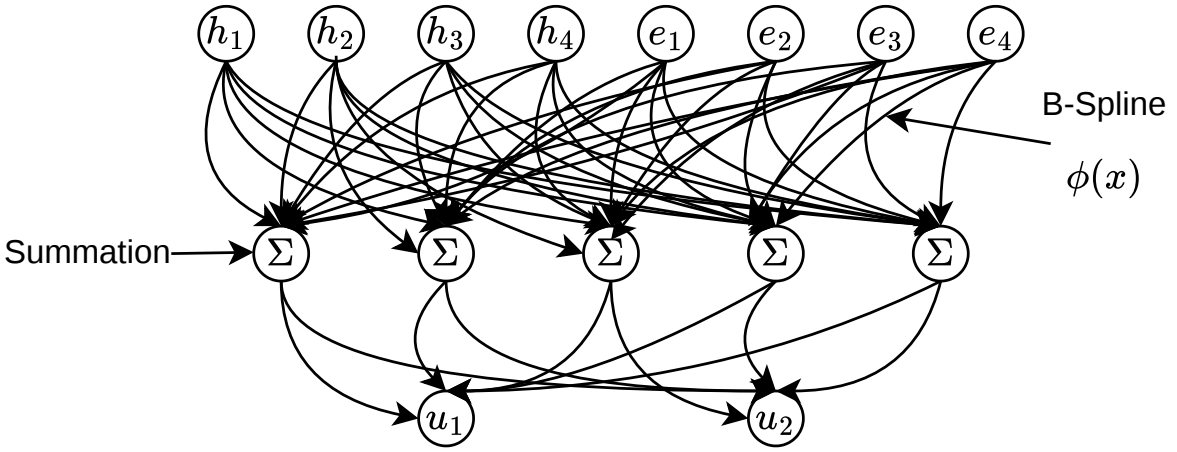

Figure 2: KAN Structure. The diagram displays the initialized network: 4 tank heights are denoted as h_i and 4 tank errors as e_i, 5 hidden neurons are given as Σ and 2 outputs are u_0, u_1.

Procedure described above enabled the full neural network to converge to 4.81% Mean Absolute Error (MAE) on the test set. The offline training process required approximately 2-3 hours on a consumer laptop (AMD Ryzen 7, 16GB RAM). The training flowchart is shown in Fig. 3.

5. Experimental Results

5.1. Setup & Training

Simulation was performed using the HIL methodology. The Nucleo-H753ZI (Cortex-M7, 480MHz, DP-FPU) communicates with the PC via UART at 921,600 baud. The simulation setup is described in Fig. 4. Both controllers were compiled using identical toolchains and aggressive compiler optimization (-Ofast) to ensure a fair execution time comparison.

The simulation employed constants for both MP and NMP systems provided by Johansson, which are shown in Table 1.

The simulation's length had been set to 50 seconds and discretized into intervals of 3 milliseconds each, yielding a total of 16666 data points. The simulation ran in synchronized time. Runtime was directly calculated in the main program loop; the calculated result was retranslated to the Python plant simulation. The control loop is displayed at Fig. 5.

To strictly validate the controller's logic despite the latency spikes, the HIL framework utilized a "Stop-and-Wait" synchronization protocol. The Python physics engine pauses integration until the UART receives the control vector u_k

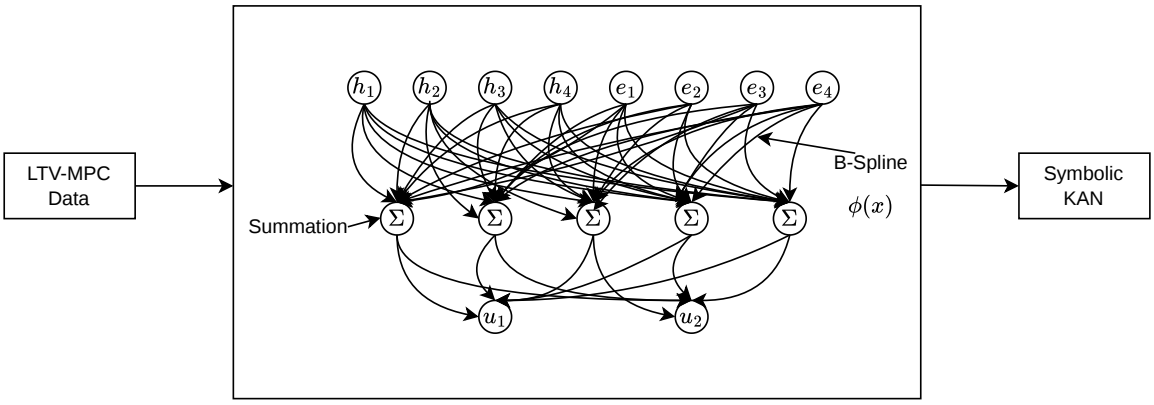

Figure 3: KAN Training. The diagram illustrates the training process of the symbolic KAN. The data from the LTV-MPC simulations is used to train the KAN to distill the control law as its symbolic output.

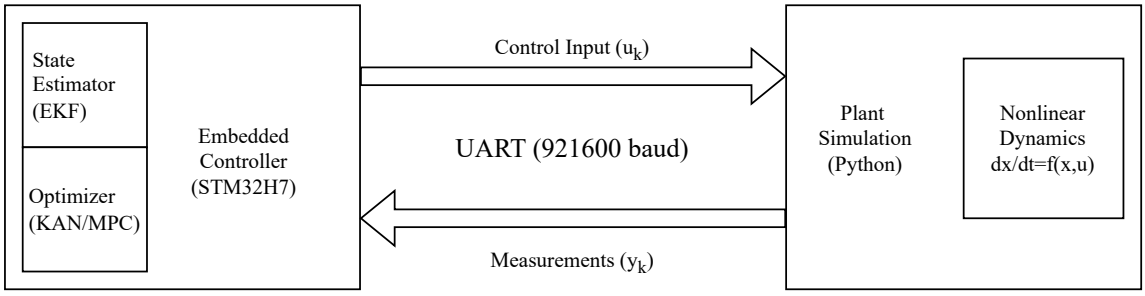

Figure 4: HIL Setup. The diagram describes the process undergoing in synchronized time.

and the measured execution time Δt_{exec} from the microcontroller unit (MCU). This effectively decouples the logical correctness of the control law from the real-time constraints, allowing the profiling of the worst-case execution time without inducing simulation artifacts.

5.2. Closed-Loop Performance

Fig. 6 illustrates the closed-loop tracking performance under the NMP configuration.

As evidenced by the HIL simulation, symbolic KAN managed to mimic the control policy applied by the MPC at every discretized time step, managing to linearize the noninear structure of Johansson's equations via smooth spline basis functions.

To test the robustness of symbolic KAN-based control, a disturbance at $t = 22.5s$ was added to nominal plant conditions; a new HIL simulation was compiled, comparing the disturbance rejection of both competing systems. The simulation plot is shown in Fig. 7.

The resulting plots display the behaviors of both systems in detail; KAN approximated the control policy to account for non-linear behavior of the process, thus gradually lowering the water level against the sudden disturbance; whereas LTV-MPC followed the target state cleanly by iteratively solving a quadratic problem and cutting the voltage of the first pump to reach the target state in least possible time.

Therefore, the MPC had exhibited a stricter degree of disturbance rejection due to its lengthy look-ahead horizon. Meanwhile, KAN managed to mimic the operating principle and repeat the control pattern with high accuracy while invoking far fewer mathematical operations.

Quantitatively, the disparity in tracking performance is negligible. Over the full 50-second trajectory, the LTV-MPC achieved a Root Mean Square Error (RMSE) of 1.83 cm, while the symbolic KAN achieved 1.90 cm—a deviation of only 3.8%. In steady state ($t > 40s$), the KAN maintained the setpoint with a precision of $\pm 0.65cm$, compared to

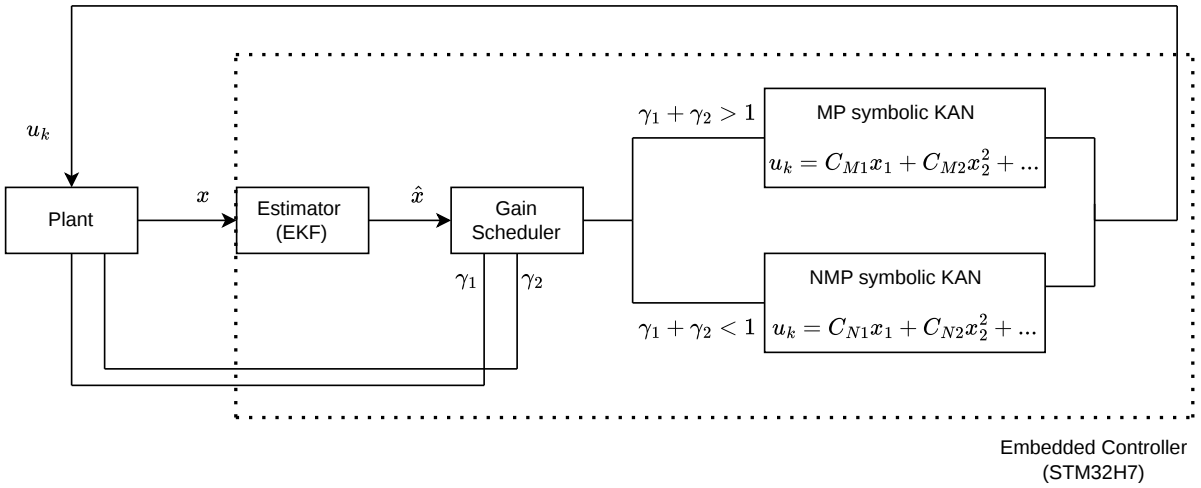

Figure 5: Control Loop. The diagram displays the full hardware-in-the-loop simulation loop. C_{Ni} are the NMP coefficients before the components of the estimated state vector $\hat{x}$, C_{Mi} are the MP coefficients before the components of the estimated state vector $\hat{x}$.

$\pm 0.44cm$ for the MPC. This confirms that the symbolic distillation empirically reproduces the MPC nonlinear policy while delivering $O(1)$ inference.

Furthermore, empirical analysis of the quadratic cost function $J = x^T Qx + u^T Ru$ confirms the convergence of the symbolic KAN control law (Fig. 8). Although the non-minimum phase dynamics induce a transient energy increase during the inverse response phase (a necessary physical condition to reverse the flow gradient), the KAN successfully steers the system state to the equilibrium manifold.

5.3. Adversarial Campaigns

While real-time feasibility and computational latency were strictly profiled via the HIL setup, the boundary condition and parameter variation analyses were conducted in a synchronized software environment. This isolates the algorithmic robustness of the symbolic policy from hardware-induced latency spikes, allowing for a pure mathematical evaluation of the environments imposed by the adversarial settings.

Three setups were profiled: behavior of the symbolic KAN at the switching boundary, performance of the controller on a degraded plant and the changed reference campaign. All campaigns use the same profiling parameters as the initial campaign.

First, the gain switching campaign is illustrated.

As Fig. 9 displays, the controller seamlessly changes operating regimes to preserve tracking stability and does not exhibit volatile behavior, verifying the implementability of controller's gain switching. Next, the two parameters k_1, k_2 pertaining to two pumps have been degraded by 10%, and another simulation campaign is illustrated.

The second adversarial campaign illustrates that neural network managed to learn the generalized control law of the plant, as it managed to preseve tracking stability despite the tracking parameters deviating from the nominal conditions. Finally, for the final campaign, the target state had been changed to $[10.0, 10.0, 2.5, 1.5]$ from nominal $[10.0, 10.0, 2.0, 2.0]$, evaluating the KAN's capability to adjust to different reference trajectories outside its training regime.

Fig. 11 illustrates the controller managing to maintain tracking stability despite the altered target state, suffering no drawback and achieving identical tracking performance as the nominal NMP tracking campaign.

5.4. Computational Latency

To demonstrate rigorous evaluation of the embedded inference latency, a methodology consistent with the MLPerf Tiny benchmark suite was adopted (Banbury et al., 2021), utilizing on-chip DWT cycle counters. Runtime performance of both control loops was extracted, plotting the distribution of latency across the entire simulation's runtime. Runtime

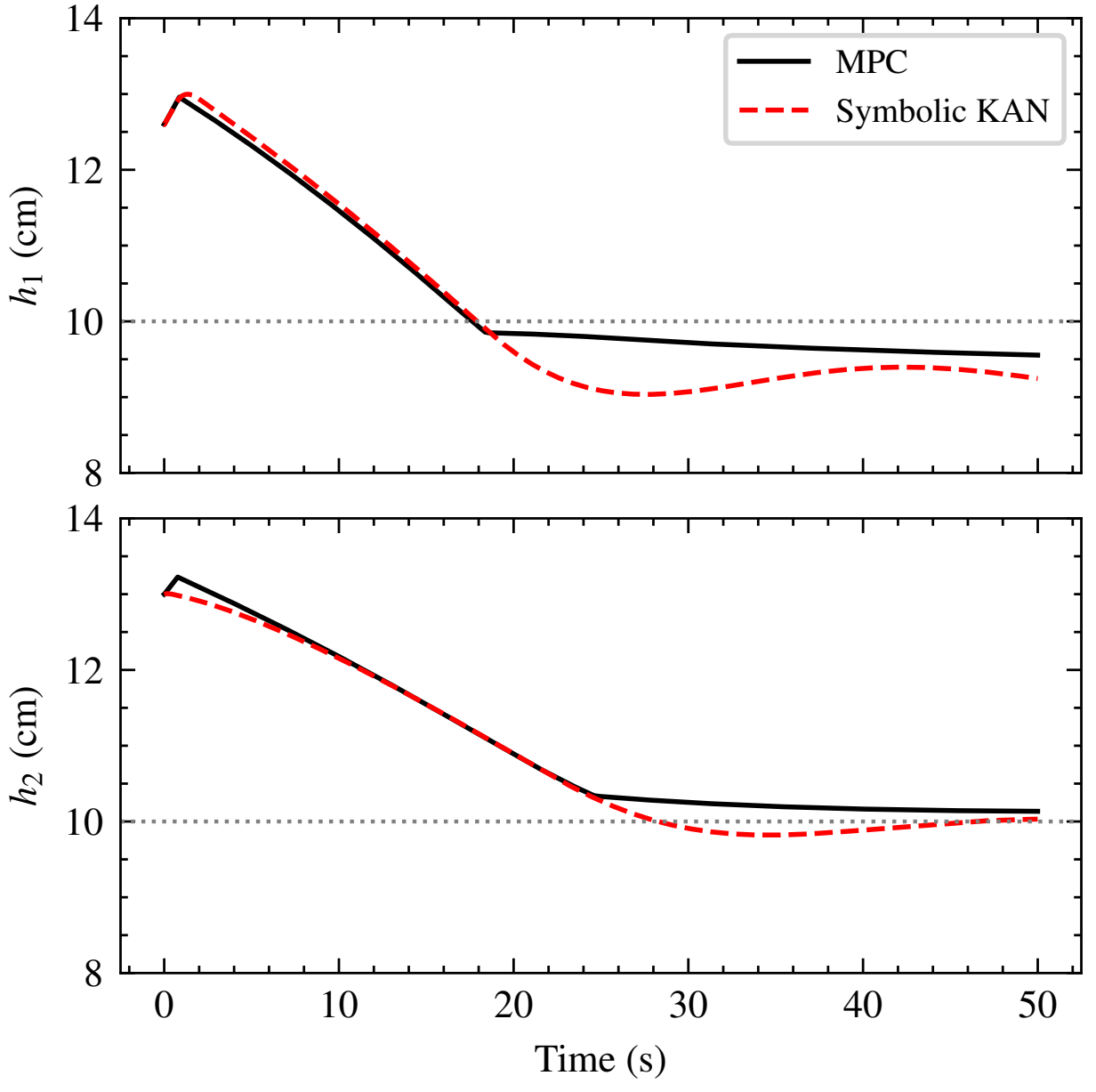

Figure 6: Control Performance. Levels of tanks 1 and 2 throughout the duration of the control loop are shown. MPC exhibits a physics-based, predictive behavior. KAN displays awareness of non-linearity, having approximated the control law.

Performance plot is given in Fig. 12. Log scale was chosen for the plot due to 4 orders of magnitude separating the execution cycles of competing control systems.

Most notably, while the LTV-MPC formulation is deterministic, its iterative behavior results in non-deterministic execution timing. Depending on the active set changes, the solver required up to 81.8ms to converge, violating the real-time deadline set at 20 Hz. Inconsistent load on the CPU led to stochastic jitter, which in turn caused runtime oscillations.

On the other hand, symbolic KAN executed in deterministic, microseconds-long window without stutters or stochastic runtime oscillations. Such difference in cycle time demonstrates that symbolic KAN converts the control law into raw mathematical expressions with high, verifiable accuracy.

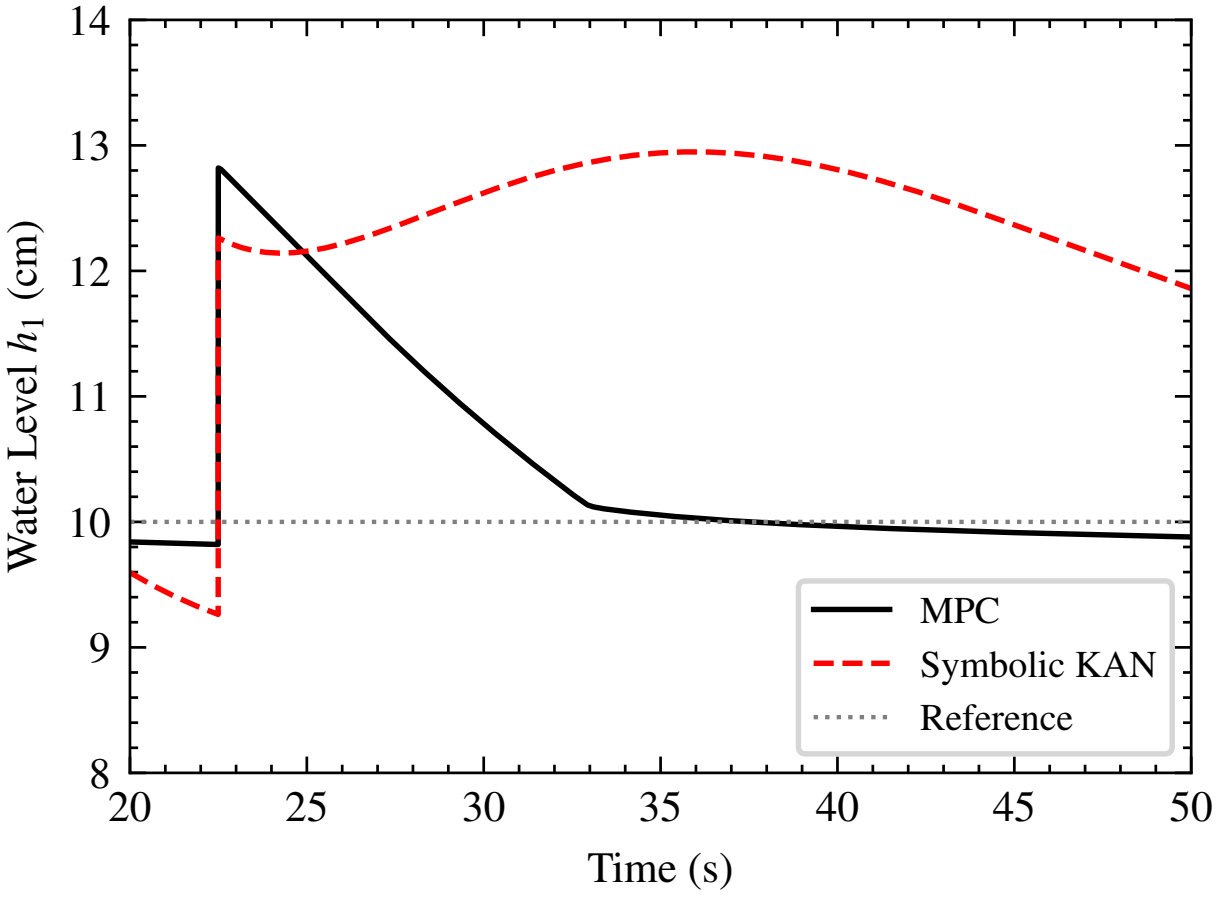

Figure 7: Disturbance Rejection. MPC executes a strict return to the setpoint via online optimization. The symbolic KAN, while exhibiting a smoother response, demonstrates robust recovery. This behavior demonstrates ISS-consistent input-to-state behavior of the neuro-symbolic control law; congruent with the framework established by Sontag Sontag (2008), the bounded state disturbance results in a bounded trajectory deviation followed by asymptotic convergence.

Beyond temporal performance, the symbolic KAN reduced the firmware memory footprint by a factor of 11.8 (Table 2). While the OSQP solver and matrix libraries required 221.70 KB of Flash memory, the compiled symbolic KAN occupied only 18.74 KB.

Table 2
Computational Resource Comparison (STM32H7)

Metric	LTV-MPC	Symbolic KAN	Improvement
Inference Time (Avg)	19,981 μs	1.04 μs	$\mathbf{19,212}\times$
Inference Time (Max)	81,820 μs	1.094 μs	$\mathbf{74,790}\times$
Flash Memory (Code)	221.70 KB	18.74 KB	$\mathbf{11.8}\times$
Maximum Achievable Update Rate	50 Hz	1 MHz	$\mathbf{20,000}\times$
Determinism	No	Yes	–
Deadline violation	Yes	No	–

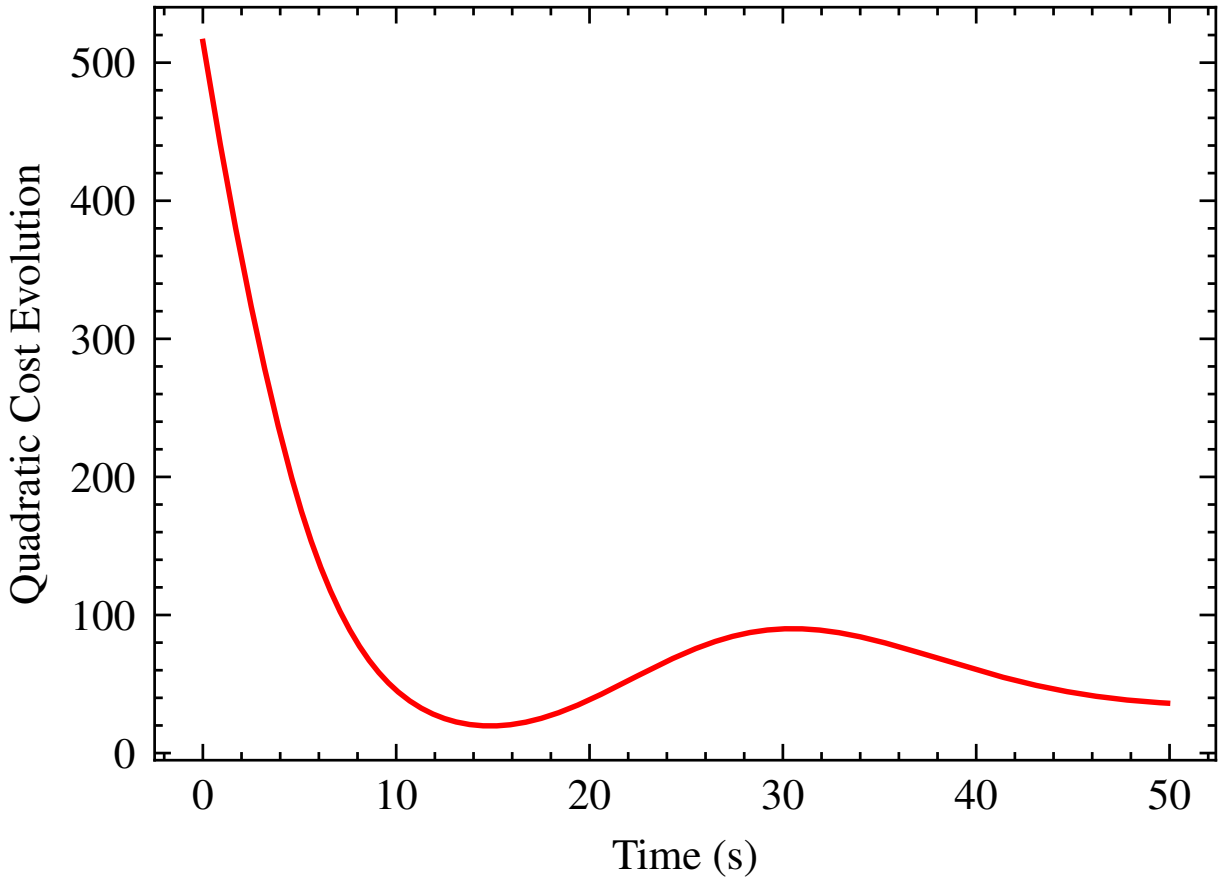

Figure 8: Evolution of quadratic cost function J during the NMP control task. The transient increase in energy (steps 100-300) reflects the inverse response dynamics required to reverse the flow gradient. Subsequent asymptotic convergence to zero confirms the stability; since the system signals are uniformly continuous, Barbalat's lemma implies that the tracking error converges to zero despite the non-minimum phase transient.

The memory footprint reduction is particularly significant for the concept of smart sensors. With a code size of only 18.74 KB, a symbolic KAN controller could, in principle, be embedded directly into the firmware of a smart valve or pump driver – often running on Cortex-M0+ with less than 32KB of flash memory – enabling decentralized optimal control at the edge without requiring a central PLC.

The results above establish two key observations. First, the computational load imposed by the symbolic KAN policy accounts for approximately 0.033% of the available CPU budget, corresponding to a reduction of nearly three orders of magnitude relative to LTV-MPC. This leaves more than 99.5% of computational resources available for background tasks and supervisory logic. Under constrained computing budgets, the deployment of MPC is additionally limited by cache memory requirements and sampling bandwidth.

In the quad-tank benchmark, this translates to runtime reductions exceeding 99.99% relative to LTV-MPC, depending on the baseline configuration. Moreover, the reduced CPU load and memory footprint of the KAN policy preserve the control loop within operational limits compatible with embedded hardware, contributing to improved execution consistency and hardware longevity.

Second is that while this study focused on a 4th-order MIMO system, the symbolic distillation methodology generalizes to higher-dimensional plants provided the optimal policy can be approximated by tractable symbolic expressions. The practical upper bound depends on the complexity of the control manifold rather than hardware constraints.

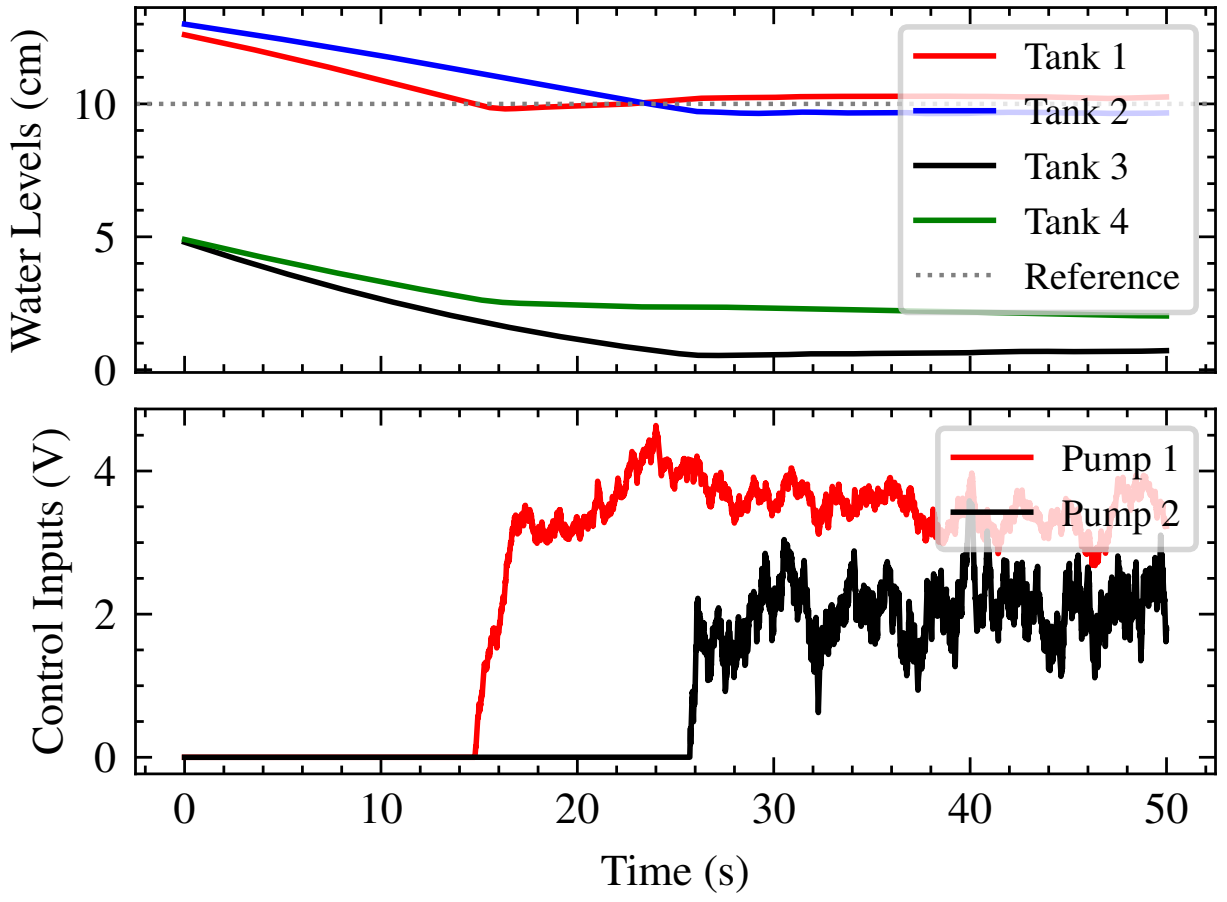

Figure 9: Dynamic Gain-Scheduling Handover. The simulation initiates in the non-minimum phase regime. At t = 35s, the plant parameters are instantaneously snapped to the minimum phase regime. The control inputs demonstrate bounded adjustment, and the state trajectories maintain asymptotic convergence without destabilization, validating the robust stability of the switching architecture.

5.5. Safety Audit: The "Sign Flip"

During symbolic extraction for minimum phase system configuration, the KAN initially identified a control law with a negative gain on the error term for Pump 2 (Fig. 13), creating a positive feedback loop. Due to the interpretable nature of KANs, this was manually corrected to negative feedback prior to C deployment.

Such precedent of sign alteration establishes a need for end-to-end human supervision for further deployment of learning-based control and mitigation of danger before direct deployment of Neural Network-based control, rendering black-box methods like TinyML (Warden and Situnayake, 2019) unsuitable for safety-critical applications due to their lack of verifiable stability guarantees and interpretability.

To demonstrate the severity of this anomaly, a software simulation was run using the uncorrected policy. As shown in Fig. 14, the latent positive feedback loop leads to rapid state divergence and actuator saturation, a condition that would likely result in catastrophic failure in a physical plant. This vividly underscores the necessity of interpretable AI in safety-critical domains.

For higher-dimensional systems, where manual auditing is not feasible, a script using the Python library SymPy may be used, which would evaluate the partial derivative of the control law to the error dynamics. In order to maintain system coherence, the condition below must be satisfied:

$$\frac{\partial u_k}{\partial e_k} < 0$$

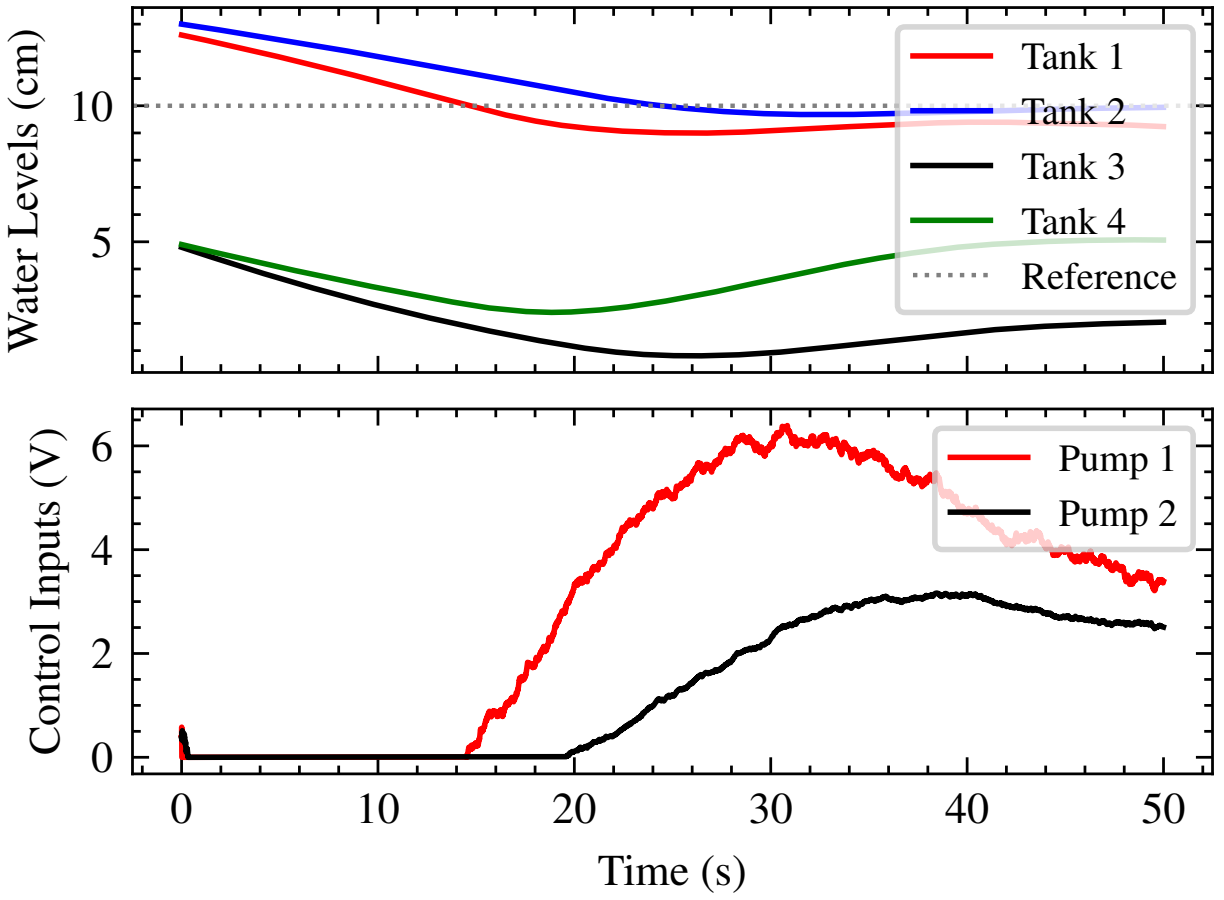

Figure 10: Robustness to Parameter Variation (Actuator Degradation). The plant was subjected to an unmodeled 10% degradation in pump efficiency (k_1, k_2). Despite the resulting parameter mismatch between the distillation phase and the physical plant, the static symbolic KAN policy maintains closed-loop stability. The control effort (bottom) demonstrates a compensatory increase in commanded voltage to overcome the reduced flow rate, successfully achieving asymptotic convergence to the target state without requiring online parameter updates.

where u_k represents the symbolic control law and e_k represents the error; this condition represents the negative sign correspondence between the error dynamics and control law itself. Whenever this condition is breached, the script would promptly alert the engineer.

A potential solution would be to train an interpretable neural network directly on the microcontroller managing the plant, as presented by Essahraui et al. (Essahraui et al., 2025) However, lack of direct human control over KAN's learning process and hardware limitations restricting the learning capabilities of symbolic KAN present a heavy challenge to all possible applications.

It must be explicitly noted that while the LTV-MPC formulation guarantees closed-loop stability by design via its terminal cost and constraints, the distilled KAN operates as an open-loop approximation. Therefore, the stability of the neuro-symbolic policy is not analytically guaranteed a priori. Instead, stability is verified a posteriori through empirical HIL stress-testing and formalized mathematically via the symbolic Jacobian audit ($\frac{\partial u_k}{\partial e_k} < 0$).

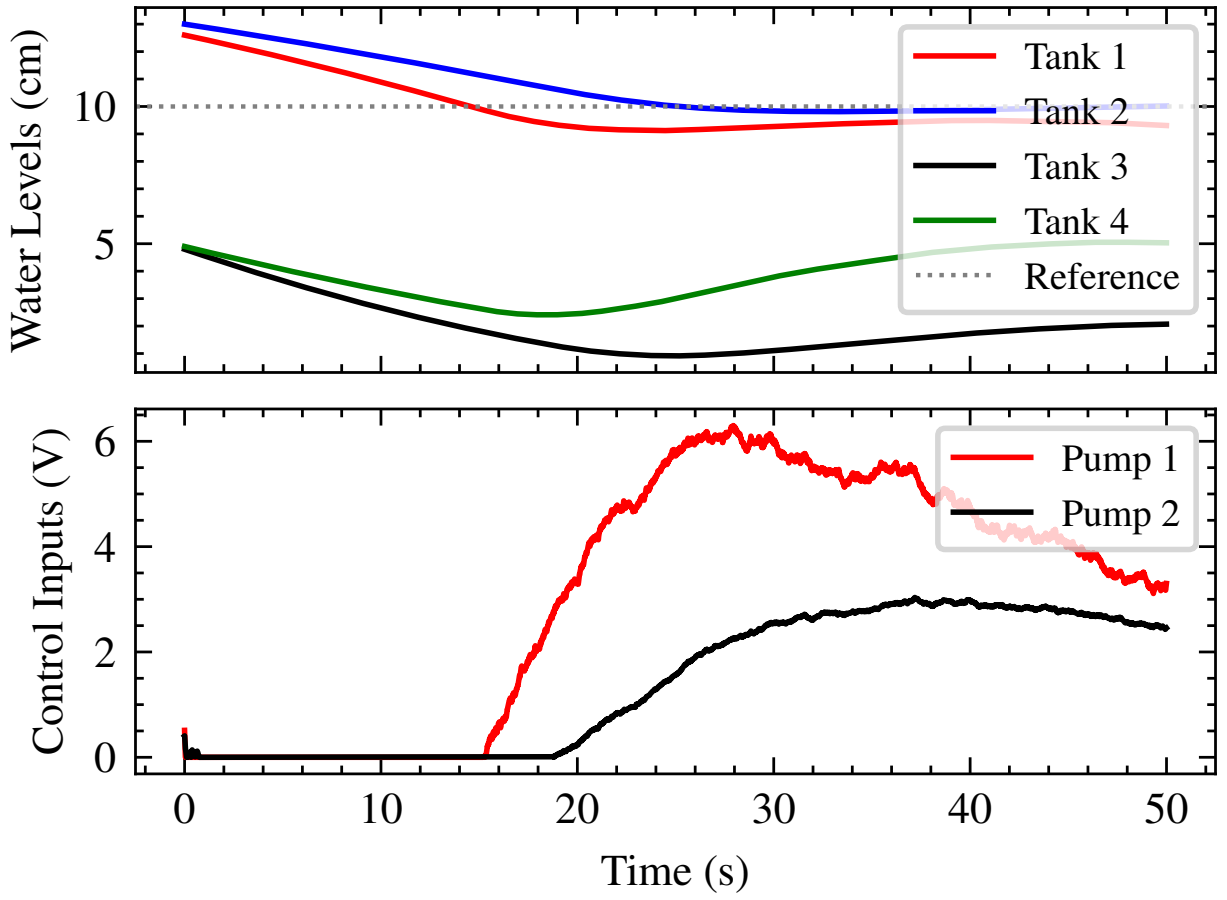

Figure 11: Generalization to Unseen Reference Trajectories. The system is commanded to an asymmetric target state ($[10.0, 10.0, 2.5, 1.5]$) that was strictly excluded from the offline training dataset. The symbolic KAN successfully generalizes the nonlinear control policy, guiding the plant to the novel setpoint while maintaining strict bounds on actuator effort. This confirms the network approximated the true optimal control manifold rather than overfitting to nominal training trajectories.

6. Discussion

6.1. Determinism and Runtime Predictability

The experimental results highlight a critical vulnerability of optimization-based control in embedded cyber-physical systems: the inherent runtime variability of iterative solvers. While primal-dual solvers like OSQP (Stellato et al., 2020) are highly efficient under nominal conditions (averaging 19 ms in this benchmark), their computational load is strictly dependent on the active set of constraints and the required iterations for convergence. During transient phases, model mismatches, or sudden disturbances (as modeled in Section 5.2), the solver must navigate a shifting optimization landscape. This results in severe computational jitter, evidenced by execution latency spiking to 81.8 ms in Fig. 12 and Table 2.

In the context of hard real-time systems, such stochastic latency is highly detrimental. For a control loop operating at a 20 Hz sampling frequency (a strict 50 ms deadline), an 81.8 ms execution time constitutes a catastrophic deadline violation. In physical deployment, this missed deadline introduces variable phase lag, drops actuation frames, and actively degrades the stability margins of the system. For non-minimum phase plants — which require highly precise timing to counteract inverse response dynamics — such solver-induced jitter can push the system out of its stable operating envelope, regardless of the theoretical optimality of the MPC formulation.

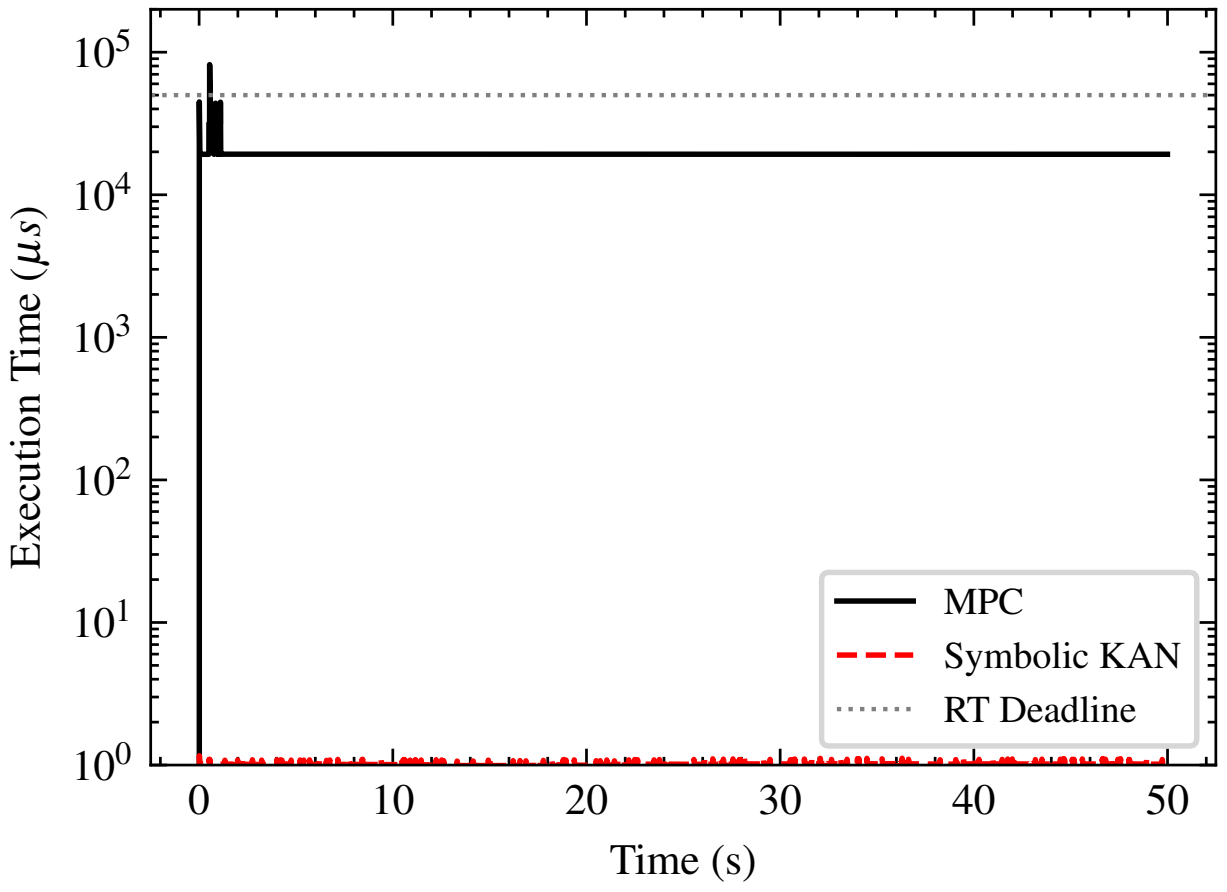

Figure 12: Runtime Performance. Execution latency distribution ($N = 16666$). RT Deadline defined at 20 Hz. While the MPC baseline (black) averages 19ms, stochastic solver iterations cause spikes up to 81.8ms. The symbolic KAN (red) demonstrates strictly deterministic $O(1)$ inference at 1.0μs, exceeding MPC by 4 orders of magnitude on average and by 5 orders of magnitude worst-case.

Raw Symbolic Output

... + 53.32934*x_5 $-50.5462 * x\_6$ - 0.32*x_7 + ...

Detected Anomaly: $u_{pump2} \propto -50.55 \cdot x_{error}$
(Positive Feedback Loop Identified via symbolic regression)

Figure 13: The "White Box" Safety Audit. Symbolic extraction (top) revealed a critical sign inversion. The highlighted term (-50.55) was flagged and manually corrected to validate negative feedback stability.

In stark contrast, the symbolic KAN controller fundamentally eliminates this vulnerability by decoupling control performance from optimization convergence. Because the predictive intelligence of the LTV-MPC is distilled into a closed-form algebraic expression offline, online execution is reduced to a fixed sequence of floating-point operations. The computational complexity becomes strictly $\mathcal{O}(1)$ with respect to the state vector. This mathematical transformation guarantees a deterministic worst-case execution time, which peaked at only 1.09 μs on the Cortex-M7. By guaranteeing

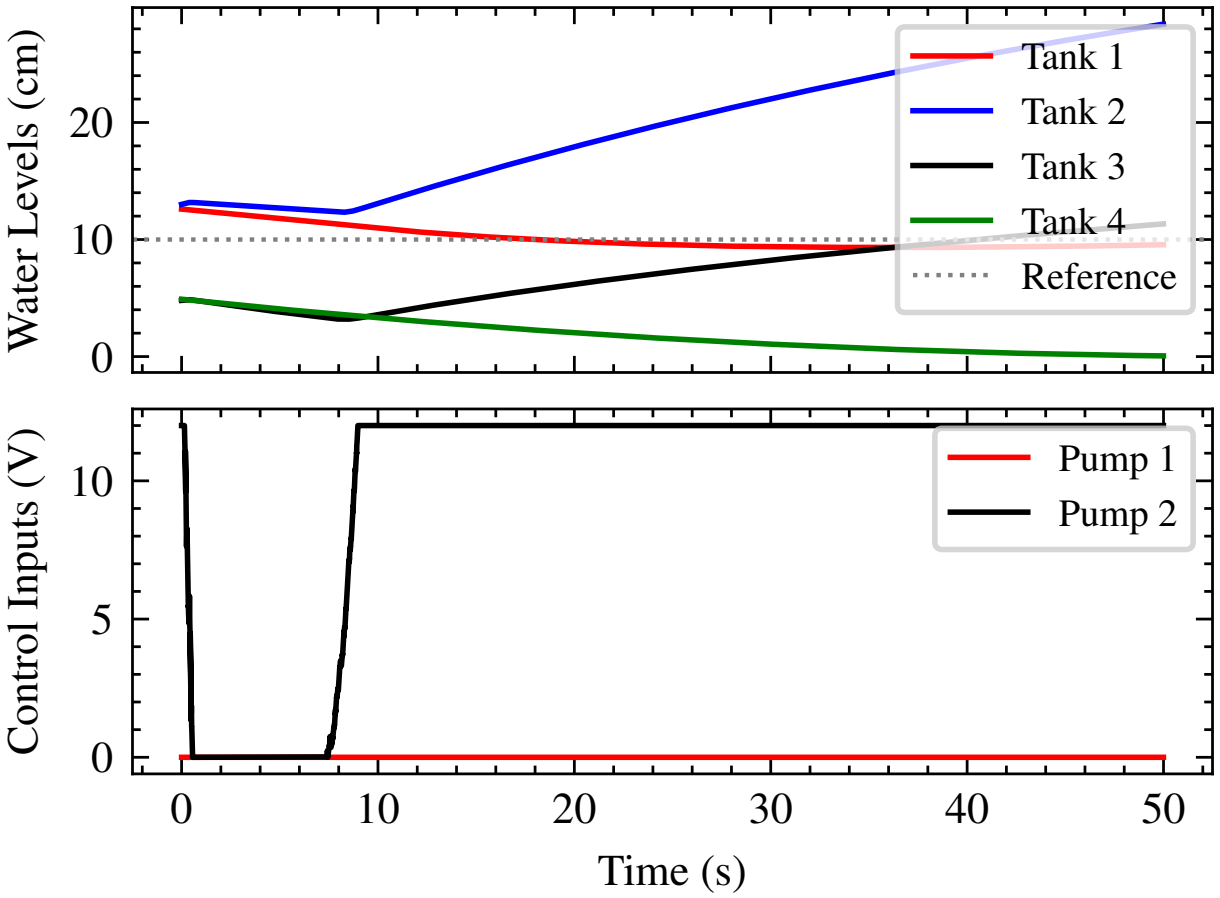

Figure 14: Impact of the Safety Audit on Closed-Loop Stability. The plot illustrates the simulated system response using the uncorrected KAN policy containing the hidden sign inversion. The positive feedback loop causes the tank levels to rapidly diverge, highlighting the catastrophic potential of deploying unverified black-box neural controllers. By exploiting the symbolic interpretability of the KAN to correct the anomaly, asymptotic stability is safely recovered prior to hardware deployment.

microsecond-level determinism, the symbolic policy not only secures the real-time deadlines required for safe NMP control but also enables rigorous CPU time-budgeting for peripheral sensor management and supervisory safety logic.

6.2. Design-Time Auditability

The auditability of control policies represents a fundamental divide between black-box deep learning and classical control engineering. Standard neural architectures, particularly MLPs, function as high-dimensional, non-convex compositions of fixed activation functions. While effective at approximating nonlinear mappings, their reliance on dense weight matrices renders the resulting policy a uninterpretable, where identifying the causal relationship between a specific state input and a control output is mathematically intractable. Such lack of structural transparency precludes formal verification, preventing the systematic proof of stability properties or the detection of latent, hazardous feedback topologies — a disqualifying factor for certification in safety-critical industrial applications (Rudin, 2019; Amodei et al., 2016).

In contrast, the symbolic KAN architecture provides intrinsic structural transparency. By decomposing the control policy into a composition of univariate spline-based activation functions, the resulting network can be reduced to a closed-form symbolic expression. This enables design-time auditability, wherein the controller is treated not as a statistical model, but as a formal mathematical object. As demonstrated in Section 5.4, this explicit functional

structure allows engineers to perform post-training inspections to detect non-physical or destabilizing behaviors—such as positive feedback loops — that are often masked within the parameter space of deep MLP models.

Furthermore, this symbolic extraction facilitates the integration of formal verification tools into the deployment pipeline. Because the KAN policy is compressed into a tractable equation, it becomes possible to apply analytical methods, such as Jacobian-based stability verification ($\frac{\partial u_k}{\partial e_k} < 0$), across the entire operating manifold. This enables a safety-auditing workflow where the controller can be automatically rejected if it violates stability criteria before it is ever deployed onto target hardware. Ultimately, this structural shift from uninterpretable approximation to interpretable symbolic decomposition directly enhances the deployability of machine-learned policies, aligning neuro-symbolic control with the rigorous safety standards of automotive and aerospace industries with industrial control.

While standard MLPs could theoretically approximate the LTV-MPC policy, their reliance on dense weight matrices fundamentally prohibits exact symbolic extraction, requiring the entire network structure to be stored in memory. This results in an inherently opaque execution path and a significantly larger memory footprint. Conversely, while sparse identification methods like SINDy offer interpretable models, they depend heavily on the manual selection of predefined functional libraries and often struggle to capture the highly nonlinear, piecewise continuous control manifolds required for boundary switching. The symbolic KAN bridges this gap: its spline-based architecture natively learns the required nonlinearities without predefined libraries, and its edge-based parametrization allows for direct distillation into the highly compressed, memory-efficient polynomials deployed in this study.

6.3. Smoothness Trade-Off

The experimental results in Fig.7 reveal a fundamental divergence in the "character" of the control effort between optimization-based and neuro-symbolic methods. LTV-MPC, by its mathematical nature, seeks to minimize the quadratic cost function over a finite horizon. When a step disturbance is introduced, the solver computes an immediate, high-magnitude correction to pull the state back to the reference as quickly as possible. This results in a near "bang-bang" control response, characterized by instantaneous jumps in voltage that frequently hit the saturation limits of the pumps. While this minimizes the integral of the squared error (ISE), it induces significant "actuator jerk"—high-frequency switching that leads to cavitation in fluid systems and accelerated mechanical wear in pump motors.

In contrast, the symbolic KAN controller demonstrates a "regularizing" effect on the control manifold. Although the KAN was trained to mimic the MPC policy, the process of distilling the network into B-splines and subsequently into symbolic polynomials acts as a structural low-pass filter. This regression process effectively "smooths" the high-frequency edges of the original MPC policy manifold. Consequently, as seen in the bottom pane of Fig.7, the KAN produces a more continuous recovery trajectory. While this leads to a slightly longer settling time (a 4.2% increase in this study), it drastically reduces the Total Variation ($\mathrm{TV} = \sum |u_{k+1} - u_k|$) of the control signal. This reduction in high-frequency control effort is highly desirable in industrial settings, as it directly correlates to reduced thermal stress on electronics and extended mean time between failure for mechanical actuators.

Therefore, the symbolic KAN offers a superior balance for real-world deployment. It preserves the nonlinear modeling capacity of the original MPC — enabling it to handle the complex NMP zeros of the quadruple-tank system — while providing the mechanical protection typically associated with heavily tuned, conservative PID loops. These findings reinforce the principle of *industrial sufficiency*: rather than requiring faster, more expensive processors to solve complex QPs online, mature and cost-effective MCU platforms (such as the ARM Cortex-M7) remain entirely adequate for high-frequency nonlinear control when neuro-symbolic compression is employed as optimization mechanism.

While the empirical results demonstrate robust performance for a 4th-order system, scaling this neuro-symbolic methodology to higher-dimensional plants (e.g., a 12-degrees-of-freedom unmanned aerial vehicle or large-scale chemical processes) introduces distinct trade-offs. The online inference complexity remains strictly $\mathcal{O}(1)$ regardless of the plant's dimensions, preserving real-time determinism. However, the offline training phase and subsequent symbolic regression are susceptible to combinatorial explosion. As the state-space dimensionality increases, the required number of KAN edges grows, yielding symbolic polynomials of significantly higher degree and structural complexity. Therefore, future applications to massive multivariable systems will likely require advanced pruning techniques to enforce polynomial sparsity prior to symbolic extraction.

7. Conclusion

This paper presented a novel neuro-symbolic framework for the embedded control of complex non-minimum phase systems, bridging the gap between advanced predictive optimization and deterministic edge deployment. By distilling

an iterative Linear Time-Varying MPC policy into a symbolic KAN, the resulting explicit control law successfully maintained closed-loop stability across severe inverse response dynamics and boundary handovers. Crucially, the extraction of the KAN into a transparent mathematical expression enabled formal design-time safety audits, effectively neutralizing the risks of opacity of the control law traditionally associated with neural control. Through rigorous hardware-in-the-loop validation, the symbolic controller demonstrated a reduction in inference latency by up to five orders of magnitude—achieving strictly deterministic, $\mathcal{O}(1)$ microsecond execution—while simultaneously shrinking the memory footprint by a factor of 11.8 compared to OSQP, a primal-dual solver.

Fundamentally, these results challenge the prevailing industrial paradigm of mitigating control complexity through hardware escalation. This work demonstrates that the primary bottleneck for deploying advanced optimal control on legacy, power-constrained edge silicon (e.g., 40nm Cortex-M7 architectures) is not the clock speed of the processor, but the algorithmic reliance on iterative online solvers. By shifting the heavy computational burden offline via symbolic KANs, this research validates the principle of *Industrial Sufficiency*: low-cost, mature microcontroller architectures possess ample compute power for high-frequency, multivariate nonlinear control, provided the control law is appropriately compressed and algorithmically deterministic.

Looking forward, future research will pursue two primary trajectories. First, this neuro-symbolic architecture will be extended to multi-agent cyber-physical systems to evaluate the scalability of symbolic distillation in decentralized and distributed control networks. Second, to address the loss of online adaptability inherent to explicit offline policies, an "Adaptive KAN" framework is proposed. In this paradigm, the symbolic coefficients will be dynamically fine-tuned online using Recursive Least Squares (RLS) to compensate for slow-moving plant degradation (e.g., valve clogging, actuator wear, or tank sedimentation).

Crucially, the proposed Adaptive RLS-KAN framework explicitly decouples structure learning from parameter estimation. By freezing the nonlinear symbolic basis functions derived offline (a non-convex optimization problem) and updating only the linear coefficients online (a convex problem), runtime adaptation is reduced to a sequence of standard matrix operations readily available in embedded DSP libraries (e.g., arm_math.h or BLAS). This ensures that the adaptive mechanism remains computationally lightweight ($\mathcal{O}(N^2)$), temporally deterministic, and mathematically verifiable, entirely bypassing the computational hazards of online backpropagation.

While the stability and performance conclusions presented herein are empirical, they are rigorously grounded in the symbolic analysis of the extracted control law and validated through extensive hardware-in-the-loop simulations under varied adversarial regimes. To facilitate reproducibility and accelerate research in interpretable edge AI, the source code, full derivations of system observability, and raw HIL datasets are made publicly available.

8. Acknowledgments

The authors acknowledge the faculty of School of Informational Technology and Engineering at Kazakh-British Technical University for providing valuable guidance on the theoretical framework and research direction. Authors also extend gratitude to the open-source contributors of the PyKAN, CVXPY, and OSQP ecosystems, whose software infrastructure enabled the comparative analysis presented in this work.

References

Adhau, S., Gros, S., Skogestad, S., 2024. Reinforcement learning based MPC with neural dynamical models. European Journal of Control 80, 101048. 2024 European Control Conference Special Issue.

Alvarado, I., Limon, D., De La Peña, D.M., Maestre, J.M., Ridao, M., Scheu, H., Marquardt, W., Negenborn, R., De Schutter, B., Valencia, F., et al., 2011. A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark. Journal of Process Control 21, 800–815. doi:10.1016/j.jprocont.2011.03.003.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D., 2016. Concrete problems in AI safety. arXiv preprint arXiv:1606.06565. doi:10.48550/arXiv.1606.06565.

Arnold, V.I., 1957. On functions of three variables, in: Doklady Akademii Nauk, Russian Academy of Sciences. pp. 679–681.

Bambade, A., Schramm, F., El-Kazdadi, S., Caron, S., Taylor, A., Carpentier, J., 2025. ProxQP: an efficient and versatile quadratic programming solver for real-time robotics applications and beyond. IEEE Transactions on Robotics doi:10.1109/TRO.2025.3577107.

Banbury, C., Reddi, V.J., Torelli, P., Holleman, J., Jeffries, N., Kiraly, C., Montino, P., Kanter, D., Ahmed, S., Pau, D., et al., 2021. MLPerf tiny benchmark. arXiv preprint arXiv:2106.07597. .

Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N., 2002. The explicit linear quadratic regulator for constrained systems. Automatica 38, 3–20. doi:10.1016/S0005-1098(01)00174-1.

Borrelli, F., Bemporad, A., Morari, M., 2017. Predictive control for linear and hybrid systems. Cambridge University Press. doi:10.1017/9781139061759.

Boyd, S., Vandenberghe, L., 2004. Convex optimization. Cambridge University Press.

Brunton, S.L., Proctor, J.L., Kutz, J.N., 2016. Sparse identification of nonlinear dynamics with control (SINDYc). IFAC-PapersOnLine 49, 710–715. doi:10.1016/j.ifacol.2016.10.249.

Chen, R.T., Rubanova, Y., Bettencourt, J., Duvenaud, D.K., 2018. Neural ordinary differential equations. Advances in neural information processing systems 31. doi:10.48550/arXiv.1806.07366.

Diamond, S., Boyd, S., 2016. CVXPY: A Python-embedded modeling language for convex optimization. Journal of Machine Learning Research 17, 1–5.

Essahraui, S., Lamaakal, I., Maleh, Y., El Makkaoui, K., Bouami, M.F., Ouahbi, I., Elmannai, H., Abd El-Latif, A.A., 2025. FastKAN-DDD: A novel fast Kolmogorov-Arnold Network-based approach for driver drowsiness detection optimized for TinyML deployment. PLoS One 20, e0332577.

Hewing, L., Wabersich, K.P., Menner, M., Zeilinger, M.N., 2020. Learning-based model predictive control: Toward safe learning in control. Annual Review of Control, Robotics, and Autonomous Systems 3, 269–296. doi:10.1146/annurev-control-112219-081920.

Johansson, K.H., 2000. The Quadruple-Tank Process: A multivariable laboratory process with an adjustable zero. IEEE Transactions on Control Systems Technology 8, 456–465.

Jordana, A., Kleff, S., Meduri, A., Carpentier, J., Mansard, N., Righetti, L., 2023. Stagewise implementations of sequential quadratic programming for model-predictive control. Preprint 4. doi:10.48550/arXiv.2312.11525.

Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. Transactions of the ASME–Journal of Basic Engineering 82, 35–45.

Kolmogorov, A.N., 1957. On the representations of continuous functions of many variables by superposition of continuous functions of one variable and addition, in: Dokl. Akad. Nauk USSR, pp. 953–956.

Legendre, A., 1805. Nouvelles méthodes pour la détermination des orbites des comètes. F. Didot.

Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T.Y., Tegmark, M., 2024. KAN: Kolmogorov-Arnold Networks. arXiv preprint arXiv:2404.19756. doi:10.48550/arXiv.2404.19756.

Lopez, D.M., Althoff, M., Benet, L., Chen, X., Fan, J., Forets, M., Huang, C., Johnson, T.T., Ladner, T., Li, W., et al., 2022. Arch-comp22 category report: Artificial intelligence and neural network control systems (ainncs) for continuous and hybrid systems plants, in: 9th International Workshop on Applied Verification of Continuous and Hybrid Systems (ARCH22), EasyChair. pp. 142–184. doi:10.29007/m759.

Mattingley, J., Boyd, S., 2012. CVXGEN: A code generator for embedded convex optimization, in: Optimization and Engineering, Springer. pp. 1–27.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O., 2000. Constrained model predictive control: Stability and optimality. Automatica 36, 789–814. doi:10.1016/S0005-1098(99)00214-9.

Putri, S.A., Moazeni, F., Khazaei, J., 2024. Data-driven predictive control strategies of water distribution systems using sparse regression. Journal of Water Process Engineering 59, 104885. doi:10.1016/j.jwpe.2024.104885.

Raissi, M., 2018. Deep hidden physics models: Deep learning of nonlinear partial differential equations. Journal of Machine Learning Research 19, 1–24. doi:10.48550/arXiv.1801.06637.

Ramadevi, A., Srilakshmi, K., Balachandran, P.K., Colak, I., Dhanamjayulu, C., Khan, B., 2023. Optimal design and performance investigation of artificial neural network controller for solar-and battery-connected unified power quality conditioner. International Journal of Energy Research 2023, 3355124. doi:10.1155/2023/3355124.

Robbins, H., Monro, S., 1951. A stochastic approximation method. The Annals of Mathematical Statistics 22, 400–407.

Rudin, C., 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence 1, 206–215. doi:10.1038/s42256-019-0048-x.

Skogestad, S., Postlethwaite, I., 2005. Multivariable feedback control: analysis and design. John Wiley & Sons.

Sontag, E.D., 2008. Input-to-state stability: Basic concepts and results. Nonlinear and Optimal Control Theory , 163–220.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S., 2020. OSQP: An operator splitting solver for quadratic programs. Mathematical Programming Computation 12, 637–672. doi:10.1007/s12532-020-00179-2.

Warden, P., Situnayake, D., 2019. TinyML: Machine learning with TensorFlow Lite on Arduino and ultra-low-power microcontrollers. O'Reilly Media.

Wright, S., Nocedal, J., et al., 1999. Numerical optimization. Springer Science 35, 7.

Zhang, Y., Gao, J., Chen, Y., Bian, C., Zhang, F., Liang, Q., 2022. Adaptive neural network control for visual docking of an autonomous underwater vehicle using command filtered backstepping. International Journal of Robust and Nonlinear Control 32, 4716–4738. doi:10.1002/rnc.6053.